

Time Management Update

AMG-16

December 17, 1996

Richard Fujimoto

Georgia Institute of Technology

Time Management Update

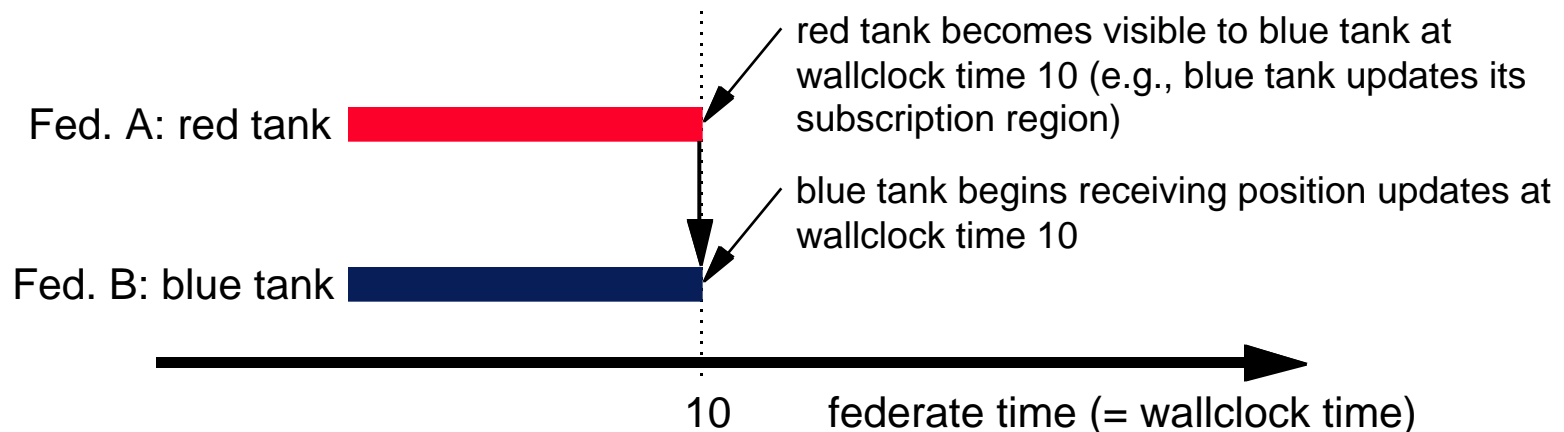
- Joint meeting with Data Distribution and Management (DDM) Group, 12/16/96
- Time Management Topics
 - Synchronized Data Distribution & Management (status report)
 - Zero lookahead and repeatable executions (changes to support zero lookahead, while maintaining repeatability)

Synchronized DDM: Problem Description

DDM: For each state update, which federates receive messages?

- example: a federate that can “see” a tank should be notified of the tank’s position updates
- set of destinations for attribute updates changes during the execution

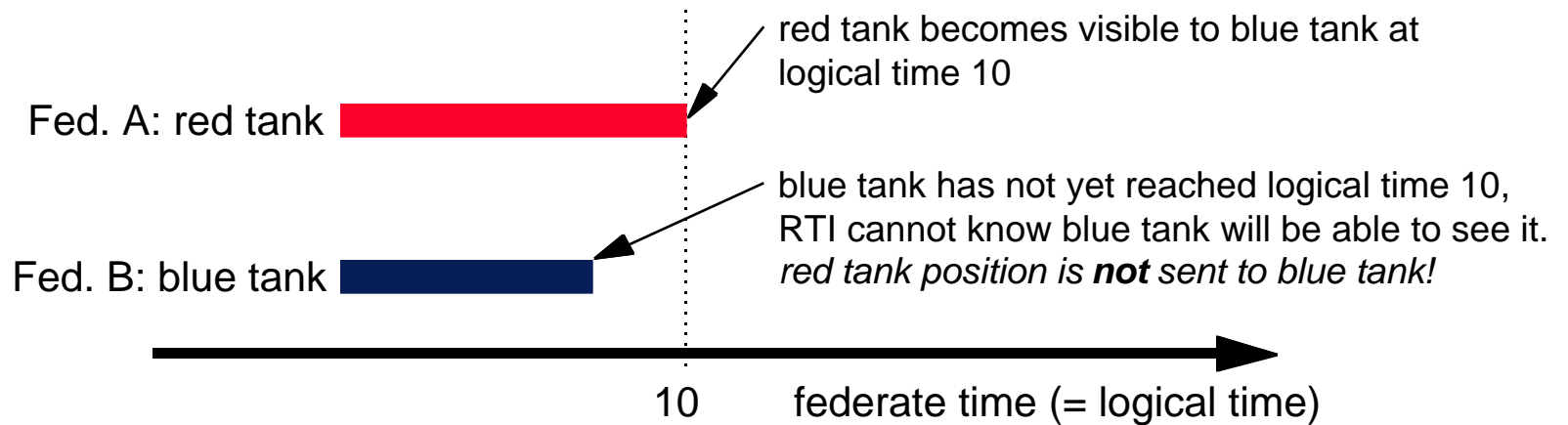
Example: Two tanks (red and blue) approaching each other in a DIS-style federation



Currently: DDM changes take effect everywhere at an instant in wallclock time

- federate time same as wallclock time (modulo offset, scale factor)
- all federates at (approx.) same federate time at any instant during the execution
- DDM changes (who receives messages?) take effect at an instant in wallclock time
- implementation: *immediately* realize changes (e.g., modify multicast groups) for all affected federates, ideally in a single instant of wallclock time
- currently supported in baseline HLA

Synchronized DDM: Problem Description (cont.)

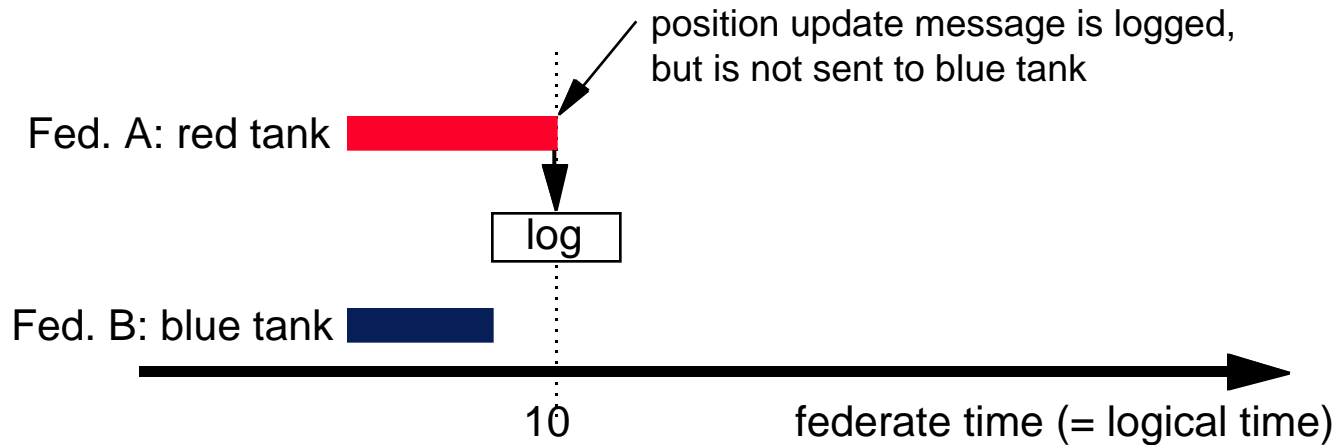


Logical time federation: DDM semantics based on logical time

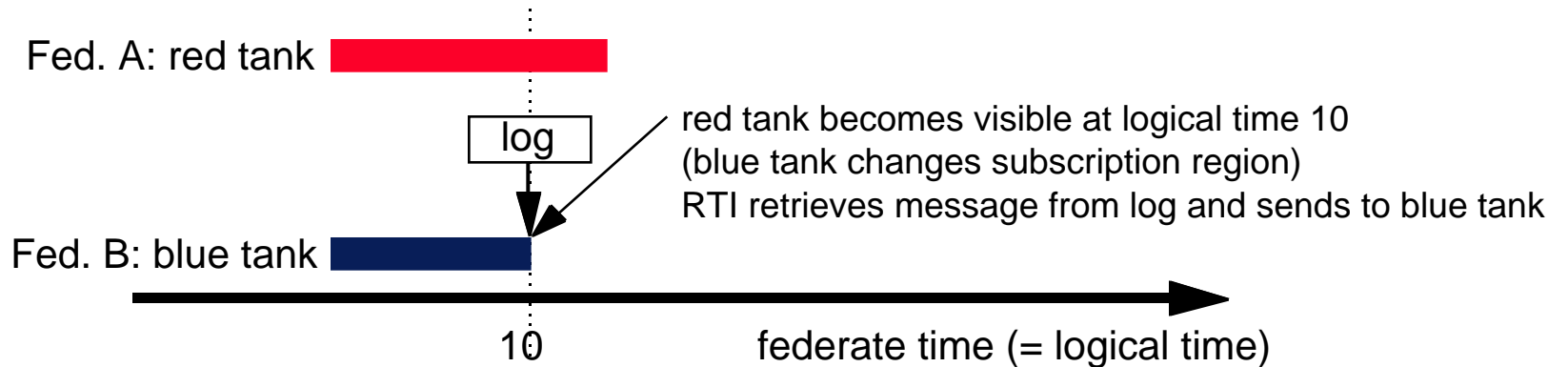
- different federates are at *different* logical times at any instant during the execution
- DDM changes take effect at a specific instant in logical time
- unless precautions are taken, a federate may not receive messages it should have received (or it may receive messages it shouldn't have received)
- logical time based DDM not currently supported in baseline HLA

A Solution Approach

- RTI maintains log of messages, sends them to federates as needed
- RTI may also retract previously sent messages



Later in the execution:



Current Status

Two additional solution approaches have been identified that do not require message logs, but place some additional constraints on federates

- require lookahead to “set up” filter changes before updates can occur (constrains when updates are generated)
- require federate to query other federates to receive “missed” information

Prototyping Efforts

- Initial prototype using logging implemented at Georgia Tech, experimentation in progress
- Other approaches still under development

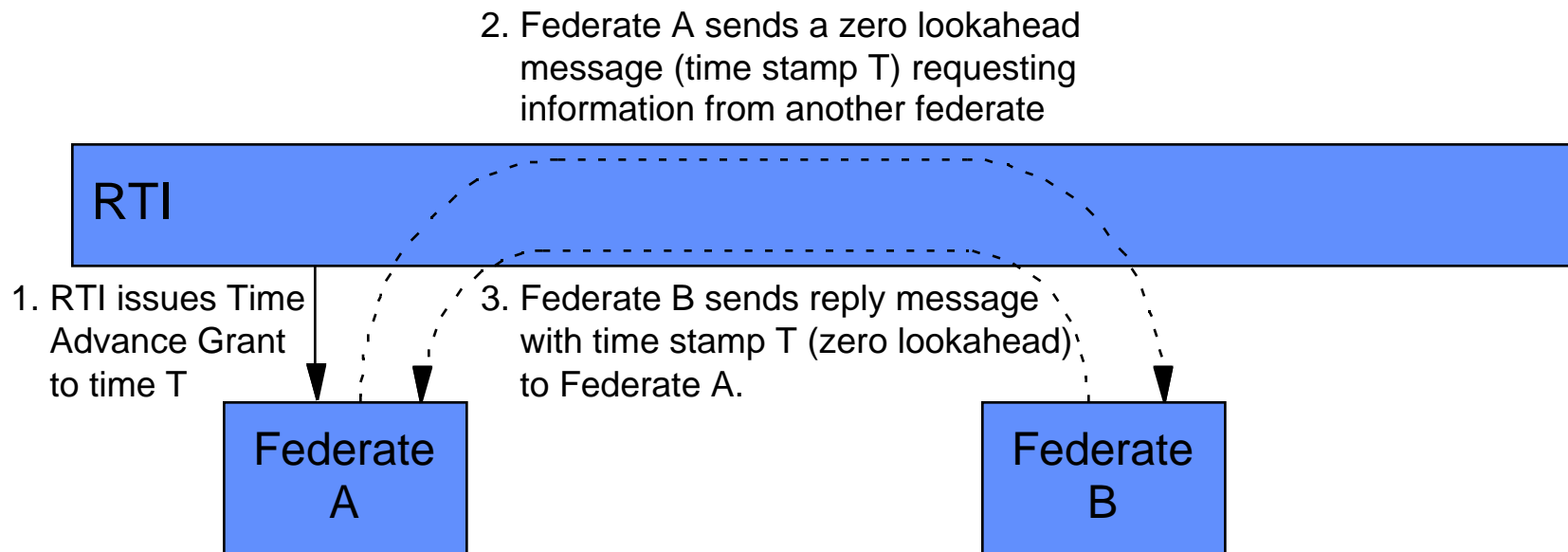
Zero Lookahead and Repeatability: Requirements

- allow zero lookahead
- repeatability: must be possible to build federations that produce the same results on each execution (analytic federations)
 - accounting purposes, facilitate debugging
 - in general, execution will be repeatable if:
 - same initial conditions and external inputs provided
 - event processing computation in each federate is repeatable
 - each federate processes events in time stamp order, and
 - events containing the same time stamp (simultaneous events) are processed in the same order in each execution
- Federation control of simultaneous events: The proper ordering of simultaneous events should be controlled by federates, not the RTI
- Avoidance of logical time creep (performance issue)
 - already addressed in Version F.0 of the RTI

Issue: repeatable, federate controlled ordering of simultaneous events

Current Approach (Repeatability): Non-Zero Lookahead

- RTI delivers all simultaneous events before issuing a *Time Advance Grant* (requires non-zero lookahead)
- Federate sorts simultaneous events containing the same time stamp in a repeatable fashion, and processes events in that order
- Above provides sufficient capability for (1) repeatable executions, and (2) federate controlled ordering of simultaneous events



The “rub”: with zero lookahead, RTI cannot guarantee delivery of all simultaneous events with time stamp T when issuing a Time Advance Grant to time T

Extensions to Support Zero Lookahead

- Existing services (Next Event Request, Time Advance Request) remain the same
 - grant to time T guarantees delivery of all messages w/ time stamp T
 - constraint: once a Time Advance Grant to time T is issued for these requests, all subsequent messages sent by the federate must have time stamp strictly greater than T.
- Two new services:
 - *Next Event Request Available* and *Time Advance Request Available*
 - Same as Next Event Request and Time Advance Request except grant to logical time T does *not* guarantee delivery of all events with time stamp T
 - Federate may send new messages with time stamp equal to T (zero lookahead)
- federate defined “ordering parameter” under consideration
- “upward compatibility” with existing RTI prototypes important

Prototyping experiments under development